МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА СИСТЕМ ІНФОРМАЦІЙНОГО ТА КІБЕРНЕТИЧНОГО ЗАХИСТУ

«На правах рукопису»
УДК 004.06

«До захисту допущено»
Завідуючий кафедрою СІКЗ
_____ к.т.н. Г.В. Шуклін
«____» _____ 2022 р.

БАКАЛАВРСЬКА АТЕСТАЦІЙНА РОБОТА

зі спеціальності 125 “Кібербезпека”

на тему: «ВПРОВАДЖЕННЯ СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ АТАК НА ОСНОВІ SURICATA»

студент групи СЗД-42 Лобанова Олександра Олександрівна _____
(підпис)

Науковий керівник: к.т.н., доцент Пепа Юрій Володимирович _____
(підпис)

Нормоконтроль: Гребенніков Асаді Болдхоядович _____
(підпис)

КИЇВ – 2022

«ЗАТВЕРДЖУЮ»

Завідувач кафедри СІКЗ

_____ к.т.н. Г.В. Шуклін

(підпис)

«_____»_____ 2022 р.

ЗАВДАННЯ

на атестаційну роботу

студенту: Лобановій Олександрі Олександрівні

1. Тема роботи: «Впровадження системи виявлення вторгнень атак на основі Suricata», затверджена наказом по університету від « » 2022 р. за № .

2. Термін здачі студентом оформленої роботи « 2 » червня 2022 р.

3. Об'єкт дослідження: системи виявлення атак.

4. Предмет дослідження: ключові характеристики IDS Suricata.

5. Мета роботи: розробити рекомендації та методичні вказівки щодо впровадження і подальшого функціонування системи раннього виявлення атак і вторгнень Suricata.

6. Перелік питань, які мають бути розроблені:

1. Актуальність проблеми впровадження системи виявлення атак.
2. Різниця між системою виявлення та системою вторгнення атак, брандмауером.
3. Зміст процесу інсталяції та конфігурації системи виявлення атак Suricata.

7. Перелік публікацій:

8. Перелік ілюстративного матеріалу. Презентація виконана на слайдах для подання за допомогою світлопроекторів та комп'ютерних засобів.

9. Дата видачі завдання «____» _____ 2022 р.

Науковий керівник _____ Пепа Ю.В.

(підпис)

Завдання прийняв до виконання _____ Лобанова О.О.

(підпис)

КАЛЕНДАРНИЙ ПЛАН

Дата видачі завдання «16» лютого 2022 р.

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз актуальності проблеми впровадження системи виявлення атак	до 29.03.22 р.	виконано
2	Аналіз архітектури IDS Suricata	до 10.04.22 р.	виконано
3	Аналіз процесу конфігурації системи виявлення атак Suricata	до 27.04.22 р.	виконано
4	Аналіз створення правил та використання вже існуючих наборів під час налаштування Suricata	до 08.05.22 р.	виконано
5	Оформлення результатів дослідження	до 16.05.22 р.	виконано
6	Підготовка демонстраційних матеріалів	до 01.06.22 р.	виконано

Студент: СЗД-42 Лобанова О.О. _____
(підпис)

Науковий керівник: к.т.н., доц. Пепа Ю.В. _____
(підпис)

Нормоконтроль: Гребенніков А.Б. _____
(підпис)

РЕФЕРАТ

Текстова частина бакалаврської роботи складається з: 37 сторінок, 12 рисунків, 2 таблиць та 9 джерел.

Об'єкт дослідження – система виявлення вторгнень як складова частина забезпечення безпеки в мережі.

Предмет дослідження – методи та засоби системи виявлення атак Suricata під час перевірки мережі на наявність вразливостей і порушень.

Мета роботи – розробити рекомендації щодо налаштування IDS Suricata для виявлення аномалій у мережі.

Методи дослідження – опрацювання літератури за даною темою, аналіз експлуатаційної документації, робота з обладнанням за софтом виробника.

Практичне значення одержаних результатів: рекомендації щодо налаштування IDS Suricata можуть бути використані у сфері забезпечення безпечної діяльності в мережі.

Галузь використання – кібербезпека.

Ключові слова: SURICATA, IDS, АНАЛІЗ МЕРЕЖІ, ЗАХИЩЕНА СИСТЕМА.

ЗМІСТ

ВСТУП

Реагування на інциденти безпеки є одним із ключових аспектів підтримки організаційної безпеки. Важливе завдання під час безпеки реагування на інцидент – це виявлення того, що інцидент стався.

Виявлення може відбуватися за допомогою звітів від кінцевих користувачів і інших зацікавлених сторін в організації шляхом аналізу, що виконується на тимчасовій основі (наприклад, ручної роботи, сценаріїв, які виявляють аномалії в журналах сервера), або це може бути здійснено за допомогою системи виявлення вторгнень (IDS).

Коли трапляються порушення в мережі, вони зазвичай викликають багато проблем і пошкоджень. Але що ще гірше, це невиявлені порушення. У цих випадках зловмисники залишаються непоміченими в цільових мережах, витягуючи якомога більше інформації.

До того часу, коли ці атаки виявляються, завдана шкода часто зростає.

Згідно з дослідженням IBM, для виявлення кібератаки в середньому потрібно 206 днів і ще 73 дні для виправлення вразливостей.

Пом’якшення загроз вимагає швидкого виявлення, щоб ідентифікувати та належним чином нейтралізувати їх, перш ніж кіберзлочинці використають будь-які вразливі місця в системі. Порушення на веб-сайті можуть підірвати надійність бренду, поставити під загрозу особисті дані третіх сторін, зупинити роботу всієї операційної системи і навіть створити юридичні проблеми для компанії.

Системи запобігання і виявлення вторгнень розроблені, щоб допомогти командам IT та NetOps подолати цю проблему. Основною метою IDPS є виявлення та блокування шкідливої поведінки, а також попередження системних адміністраторів про наявність шкідливого трафіку або вторгнення в мережу.

Отже, концепція виявлення вторгнень існує вже багато років і залишатиметься необхідною, доки зловмисники намагатимуться зламати мережі та вкрасти конфіденційні дані.

1 СИСТЕМА ВИЯВЛЕННЯ ВТОРГНЕНЬ

1.1 Система виявлення вторгнень

Система виявлення атак – це додаток або пристрій, який відстежує вхідний і вихідний мережевий трафік, постійно аналізуючи активність на предмет змін у шаблонах і попереджає адміністратора, коли виявляє незвичайну поведінку. Потім адміністратор переглядає сигнали тривоги та вживає заходів для усунення загрози.

Наприклад, IDS може перевіряти дані, що передаються мережевим трафіком, щоб побачити, чи містять вони відоме зловмисне програмне забезпечення або інший шкідливий вміст. Якщо вона виявляє загрозу такого типу, то надсилає сповіщення команді безпеки, щоб вони могли її розслідувати та усунути. Як тільки команда отримає сповіщення, вона повинна діяти швидко, щоб запобігти атаці, яка захопить систему.

Щоб переконатися, що IDS не сповільнює продуктивність мережі, ці рішення часто використовують аналізатор комутованого порту (SPAN) або порт тестового доступу (TAP) для аналізу копії вбудованого трафіку даних. Однак вони не блокують загрози після того, як вони входять в мережу, як це роблять системи запобігання вторгненням. Незалежно від того, налаштували ми фізичний пристрій чи програму IDS, система може:

1. Розпізнавати моделі атак у мережевих пакетах.
2. Відстежувати поведінку користувачів.
3. Визначати ненормальну активність трафіку.
4. Переконатися, що діяльність користувачів і системи не суперечить політикам безпеки.

Інформація із системи виявлення вторгнення також може допомогти команді безпеки:

1. Перевірити мережу на наявність вразливостей і поганих конфігурацій.
2. Оцінити цілісність критичних систем і файлів.
3. Створити ефективніші засоби контролю та реагування на інциденти.
4. Проаналізувати кількість і типи кіберзагроз, що атакують мережу.

1.2 Структура та архітектура загальної системи виявлення вторгнень

У системі виявлення вторгнень є чотири основні частини. На рис. 1.1 вони показані як сховище інцидентів, аналізатори, блок реагування та сенсори.

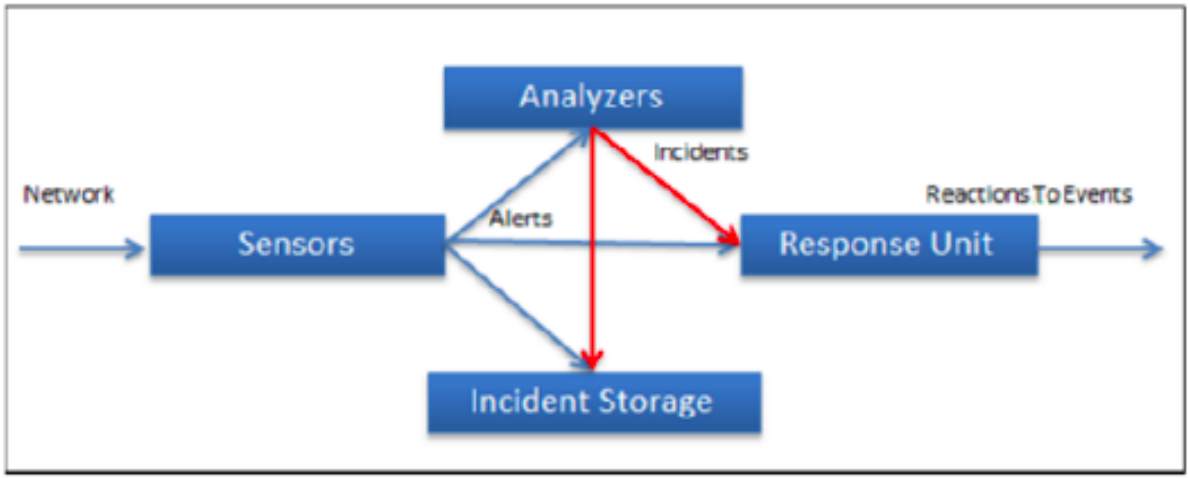


Рисунок 1.1 – Структура системи виявлення вторгнень

- Сенсори. Визначають і надсилають дані в систему.
- Аналізатори або Центральна система моніторингу. Обробляє та аналізує дані, надіслані з сенсорів.
- Компоненти бази даних і сховищ. Виконують аналіз тенденцій і зберігають IP-адресу та інформацію про зловмисника.
- Поле відповіді. Вводить інформацію з перерахованих раніше компонентів і формує певну відповідь.

Системи виявлення вторгнень завжди мають свій основний елемент – сенсор (систему аналізу), який відповідає за виявлення вторгнень.

Сенсори отримують необроблені дані з трьох основних джерел інформації:

- власна база знань IDS;

- системний журнал;

- аудиторські сліди.

Системний журнал може включати, наприклад, конфігурацію файлової системи, авторизації користувачів тощо. Ця інформація створює основу для подальшого процесу прийняття рішень. Сенсор інтегрований з компонентом, що відповідає за збір даних – генератором подій. Спосіб збирання визначається політикою генератора подій, яка визначає режим фільтрації інформації про подію.

Генератор подій (операційна система, мережа, програма) створює узгоджений з політикою набір подій, які можуть бути журналом (або аудитом) системних подій або мережевих пакетів.

Роль сенсора, як ми бачимо на рис. 1.2 полягає в тому, щоб фільтрувати інформацію та відкидати будь-які невідповідні дані, отримані з набору подій, пов’язаних із захищеною системою, таким чином виявляючи підозрілі дії. Для цього аналізатор використовує базу даних політики виявлення. Крім того, база даних містить параметри конфігурації IDS, включаючи режими зв'язку з модулем відповіді. Датчик також має власну базу даних, що містить динамічну історію потенційного комплексу.

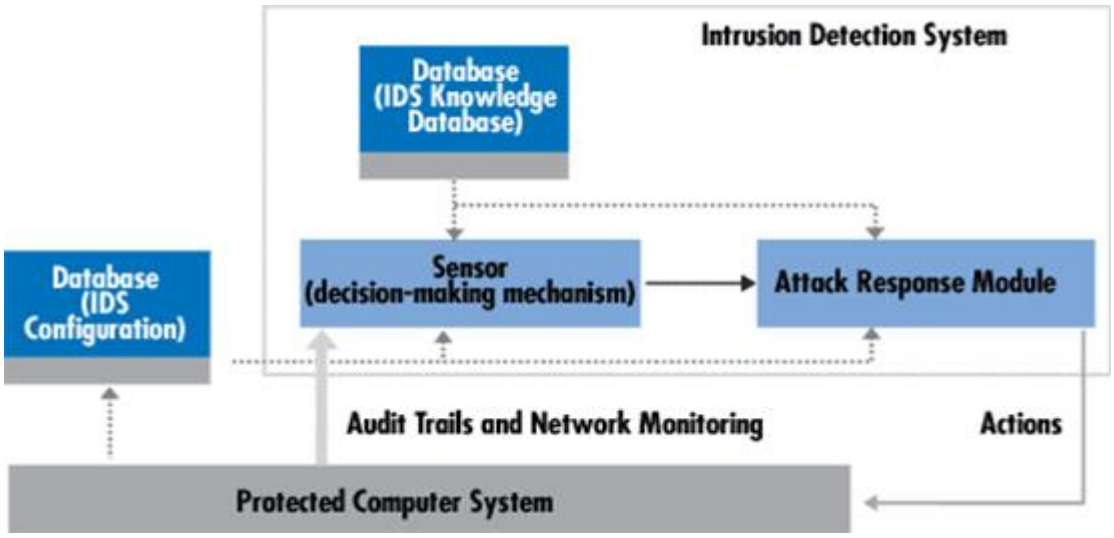


Рисунок 1.2 – Архітектура системи виявлення атак

1.3 Категоризація систем виявлення атак

Залежно від розміщення, IDS працює по-різному і може бути розділена на мережеву IDS (NIDS) , IDS на основі хосту (HIDS) і IDS на основі стека (SIDS) (рис. 1.3).

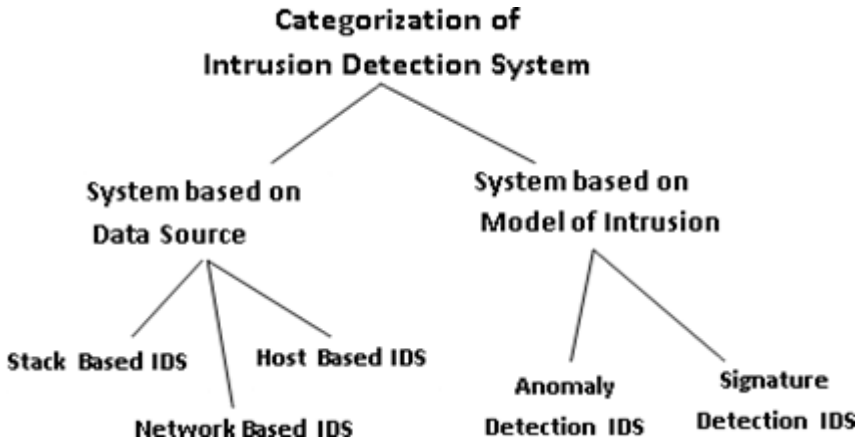


Рисунок 1.3 – Категоризація систем виявлення атак

SIDS – це новітня технологія, яка працює шляхом ретельної інтеграції зі стеком TCP/IP, дозволяючи спостерігати за пакетами, коли вони переміщуються вгору по рівнях OSI. Спостерігаючи за пакетом таким чином IDS витягує пакет зі стека до того, як операційна система або програма матиме шанс обробити пакети.

NIDS сканує величезні обсяги мережевої активності на рівні маршрутизатора та позначає підозрілі передачі, такі як підробка IP, атаки DoS, отруєння кешу ARP, пошкодження імен DNS та атаки «людина посередині».

HIDS відстежує декілька файлів журналів на хості (ядро, система, мережа, брандмауер), щоб виявити неправильне використання або вторгнення. Крім того, HIDS забезпечує цілісність критичних даних на хості шляхом перевірки контрольних сум файлів (md5 або sha1). Якщо контрольні суми не збігаються, HIDS повідомляє про це адміністратора.

1.4 Мета NIDS та HIDS

І HIDS, і NIDS (рис. 1.4) фіксують мережевий трафік і порівнюють зібрану інформацію з попередньо визначеними шаблонами, щоб виявити атаки та вразливості.

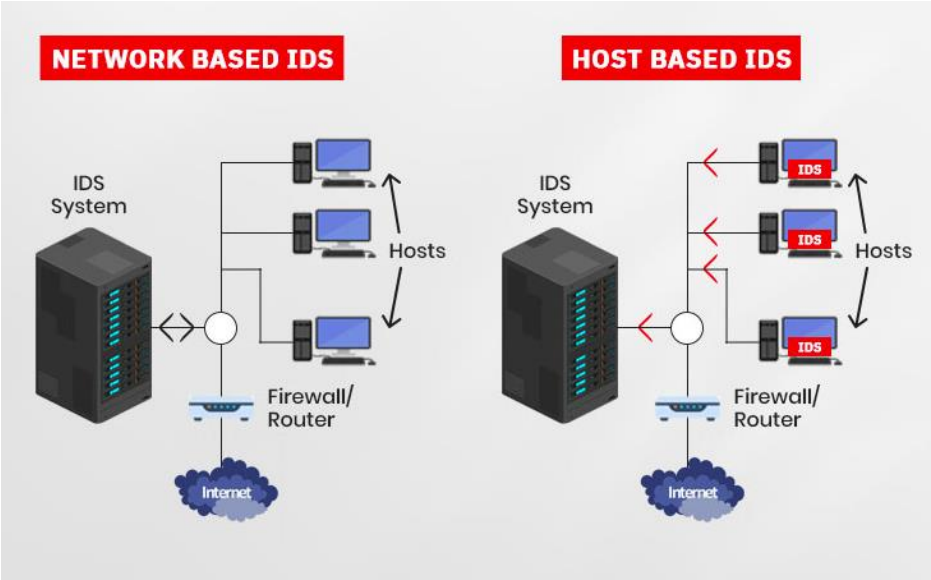


Рисунок 1.4 – Як функціонують NIDS та HIDS

1.5 Порівняння NIDS та HIDS

Основні функції HIDS досліджують конкретні дії на базі хоста, наприклад, яке програмне забезпечення було використано, до яких документів було здійснено доступ і яка інформація міститься в журналах ядра. У той же час NIDS досліджують потік даних між комп'ютерами (мережевий трафік). Таким чином, NID можуть виявити хакера, поки він не здійснить несанкціоновану атаку, тоді як HID не зрозуміють, що щось не так, поки хакер не зламав систему.

Крім місця розташування, IDS можна класифікувати за методом виявлення (рис. 1.5).

Рисунок 1.5 – Класифікація IDS за методом виявлення

На основі сигнатур – ці системи здійснюють пошук шаблонів у трафіку та порівнюють їх із базою даних підписів. Хоча системи на основі сигнатур можуть легко виявити відомі шкідливі шаблони, вони не можуть визначити нові атаки, оскільки їх шаблон недоступний у базі даних. База сигнатур повинна постійно оновлюватися, як і антивірусні бази. Цей метод виявлення дає найменшу кількість помилкових спрацьовувань.

На основі аномалій – замість того, щоб покладатися на бази даних сигнатур для запобігання відомим атакам, цей метод заснований на машинному навчанні. Спочатку IDS створює модель на основі надійної діяльності, а потім порівнює нові дії в системі з початковою моделлю. Цей метод виявлення може викликати помилкові спрацьовування, оскільки діяльність, яка спочатку не моделювалася як надійна, може спровокувати IDS.

На основі правил/політики – цей метод покладається на базу знань адміністратора, а також конструкції логічного й умовного програмування (якщо УМОВА виконай ДІЯ), як показано на рис. 1.6.

Rule Types	SNORT	Suricata	OSSEC	Rule Types	Snort	Suricata	OSSEC
bad-traffic	yes	yes	yes	attack-responses	yes	yes	yes
exploit	yes	yes	yes	oracle	yes	yes	yes
scan	yes	yes	yes	mysql	yes	yes	yes
finger	yes	yes	yes	snmp	yes	yes	yes
ftp	yes	yes	yes	smtp	yes	yes	YES
telnet	yes	yes	yes	imap	yes	yes	yes
rpc	yes	yes	yes	pop2	yes	yes	no
rservices	yes	yes	yes	pop3	yes	yes	yes
dos	yes	yes	yes	nntp	yes	yes	yes
ddos	yes	yes	yes	web-attacks	yes	yes	yes
dns	yes	yes	yes	backdoor	yes	yes	yes
tftp	yes	yes	yes	shellcode	yes	yes	yes
web-cgi	yes	yes	yes	policy	yes	yes	yes
web-coldfusion	yes	yes	yes	porn	yes	yes	yes
web-iis	yes	yes	yes	info	yes	yes	yes
web-frontpage	yes	yes	yes	icmp-info	yes	yes	yes
web-misc	yes	yes	yes	virus	yes	yes	yes
web-client	yes	yes	yes	chat	yes	yes	yes
web-php	yes	yes	yes	multimedia	yes	yes	yes
sql	yes	yes	yes	p2p	yes	yes	yes
x11	yes	yes	yes	spyware-put	yes	yes	no
ssh	yes	yes	yes	specific-threats	yes	yes	yes
icmp	yes	yes	yes	experimental	yes	yes	yes

Рисунок 1.6 – таки, які можна виявити за допомогою різних IDS

1.5 Перспективи викликів IDS

Продуктивність поточних IDS не захищає зростаючу кількість типів атак, оскільки багато поточних IDS все ще базуються на правилах, які вручну створені експертами і описують лише відомі сигнатури атак. Існує три точки зору технічних проблем у IDS на основі машинного навчання, а саме: вилучення ознак, побудова класифікатора та послідовне передбачення шаблонів.

1. Виділення ознак є основою для високоефективного виявлення вторгнень.

Якщо функції вибрано неправильно, то на кінцеву продуктивність моделей виявлення сильно вплине.

2. Побудова класифікатора.

Точність класифікації більшості існуючих методів потребує покращення, оскільки дуже важко виявити багато нових атак шляхом навчання лише обмежених даних аудиту. Використання стратегії виявлення аномалій може виявити нові атаки, але частота помилкових тривог зазвичай дуже висока, оскільки дуже добре моделювати нормальні моделі також важко. Таким чином, побудова класифікатора в IDS залишається ще одним технічним завданням для виявлення вторгнень на основі машинного навчання.

3. Послідовне передбачення шаблону.

Проблему виявлення вторгнень на основі хоста можна розглядати як проблему послідовного передбачення, оскільки важко визначити одну коротку послідовність системних викликів як нормальну чи ненормальну, а між послідовностями існують внутрішні тимчасові зв’язки. Хоча ми все ще можемо перетворити вищезазначену проблему на проблему статичної класифікації, зіставляючи весь слід процесу з вектором ознак, було показано, що динамічні методи моделювання поведінки, такі як приховані моделі Маркова (HMM), більше підходять для цього різновиду проблеми виявлення вторгнення.

1.6 Ідеальна система виявлення вторгнень

Ідеальна система виявлення вторгнень повинна вирішувати такі проблеми, незалежно від механізму, на якому вона заснована:

1. Система повинна працювати постійно без нагляду людини. Вона повинна бути достатньо надійною, щоб мати змогу працювати у фоновому режимі системи, за якою спостерігають.

2. Це не має бути "чорний ящик". Тобто внутрішня робота системи має бути оглянутою ззовні.

3. Вона повинна протистояти підривній діяльності. Система має стежити за собою, щоб переконатися, що вона не була підірвана.

4. Накладні витрати на систему мають бути мінімальними. Система, яка сповільнює роботу комп'ютера, просто не використовуватиметься.

5. Система повинна спостерігати відхилення від нормальної поведінки.

6. Кожна система має різну схему використання, і захисний механізм повинен легко адаптуватися до цих шаблонів.

7. Вона повинна мати справу зі зміною поведінки системи з часом у міру додавання нових програм. З часом профіль системи буде змінюватися.

8. Її має бути важко обдурити.

1.7 Порівняння Firewall та IDS

Firewall обмежує доступ до мережі, перевіряючи трафік і вирішуючи, які пакети мають бути дозволені. Firewall відстежує порти, які підключають мережу до Інтернету, і перевіряє пакети даних, перш ніж дозволити їм пройти. Firewall може прийняти пакет, скинути його або відхилити, повернувши відправнику (табл. 1.1).

Таблиця 1.1

Порівняння Firewall та IDS

Firewall	IDS
Firewall – це апаратне та/або програмне забезпечення, яке функціонує в мережевому середовищі для блокування несанкціонованого доступу.	IDS – це програмний або апаратний пристрій, встановлений у мережі (NIDS) або на хості (HIDS) для виявлення та повідомлення про спроби вторгнення в мережу.

Продовження таблиці 1.1

Firewall	IDS
Може блокувати несанкціонований доступ до мережі.	Повідомляє лише про вторгнення; не може його заблокувати.
Не може виявити порушення безпеки для трафіку, який не проходить через нього.	Здатна забезпечити внутрішню безпеку, збираючи інформацію з різноманітних системних і мережевих ресурсів і аналізуючи симптоми проблем безпеки.
Є найбільш помітною частиною мережі для сторонніх осіб. Отже, вразливіші для нападу першими.	Дуже важко помітити в мережі (особливо прихований режим IDS).
Не перевіряє вміст дозволеного трафіку.	IDS перевіряє загальну мережу.
Для керування не потрібна людська сила.	Для реагування на загрози IDS потрібен адміністратор (людська сила).

1.8 Порівняння IDS та IPS

IPS – це інструмент безпеки мережі (який може бути апаратним пристроєм або програмним забезпеченням), який постійно контролює мережу на наявність шкідливої діяльності та вживає заходів, щоб запобігти її, включаючи звітування, блокування або видалення, коли це сталося.

IPS зазвичай записує інформацію, пов’язану з спостережуваними подіями, повідомляє адміністраторів безпеки про важливі спостережувані події та створює звіти. Багато IPS також можуть реагувати на виявлену загрозу, намагаючись запобігти її успіху. Системи використовують різні методи реагування, які передбачають, що IPS зупиняє саму атаку, змінює середовище безпеки або змінює вміст атаки.

IPS розглядаються як доповнення до систем виявлення вторгнень (IDS), оскільки як IPS, так і IDS керують мережевим трафіком і системною діяльністю для зловмисної діяльності.

Основна відмінність між ними полягає в тому, що IDS – це система моніторингу, а IPS - це система управління.

IDS жодним чином не змінює мережеві пакети, тоді як IPS запобігає доставці пакету на основі вмісту пакета, подібно до того, як брандмауер запобігає трафіку за IP-адресою.

IDS слід розташовувати після брандмауера, тоді як IPS слід розміщувати після пристрою брандмауера в мережі (рис. 1.7).

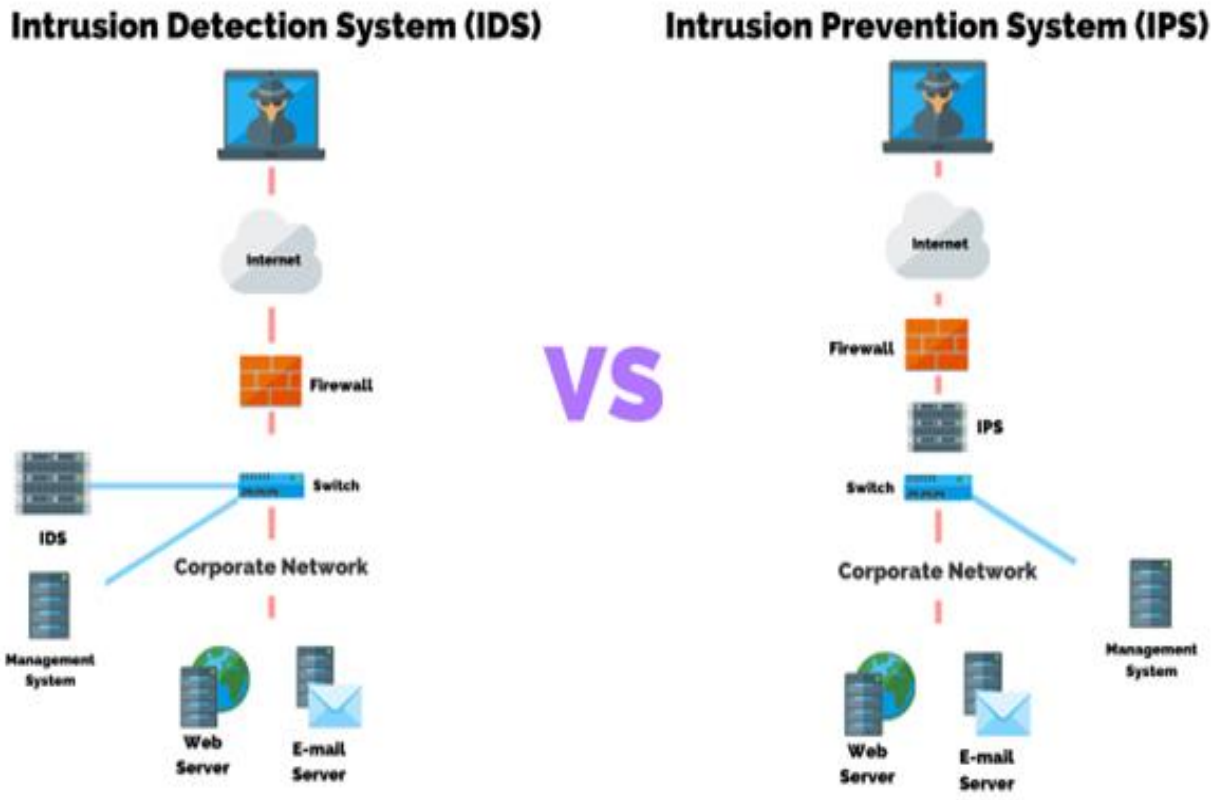


Рисунок 1.7 – Розташування IDS та IPS

Порівняємо IDS і IPS (табл. 1.2).

Таблиця 1.2

Порівняння IDS і IPS

IDS	IPS
Інструменти IDS були створені для виявлення зловмисної активності, реєстрації та надсилання сповіщень. Вони не здатні запобігти нападу. Попередження, які вони висувають, завжди вимагають втручання людини або додаткової системи безпеки.	Відповіді IPS мають в основі заздалегідь визначені критерії типів атак шляхом блокування трафіку та видалення шкідливих процесів. Інструменти IPS призводять до більшої кількості помилкових спрацьовувань, оскільки вони мають нижчі можливості виявлення, ніж IDS.

2 СИСТЕМА ВИЯВЛЕННЯ АТАК SURICATA

2.1 Характеристика системи виявлення атак Suricata

Suricata – це механізм виявлення загроз у реальному часі, який допомагає захистити вашу мережу від загроз шляхом активного моніторингу мережевого трафіку та виявлення шкідливої поведінки на основі написаних правил (рис. 2.1).

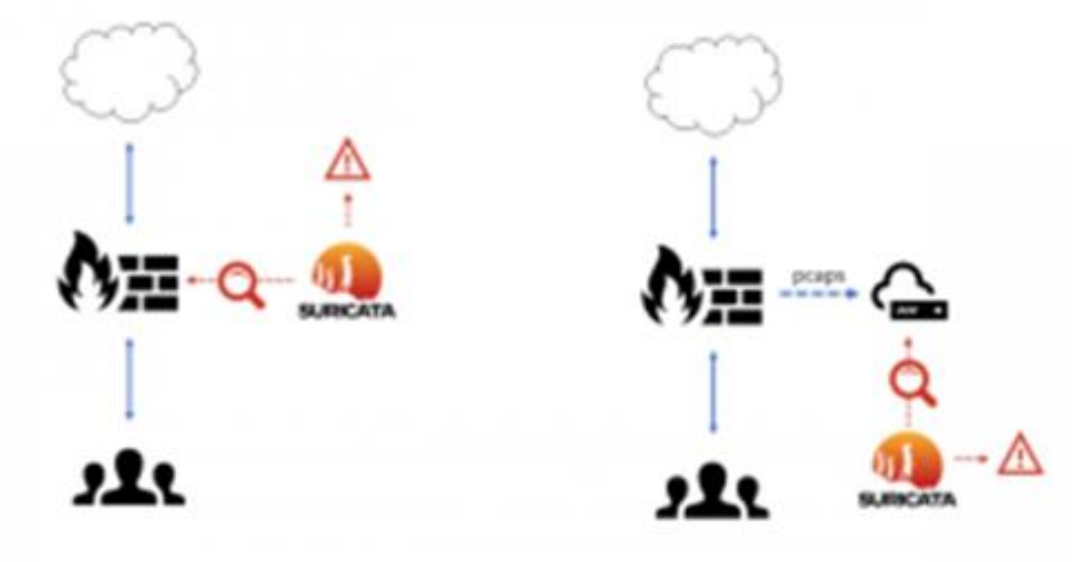


Рисунок 2.1 – Ліворуч: Suricata як моніторинг мережі; праворуч: Suricata як аналізатор пакетів

Вона може працювати в режимі моніторингу безпеки мережі, а також може бути налаштована як система запобігання вторгненню (IPS) або система виявлення вторгнень (IDS).

Проект Suricata є відкритим вихідним кодом і виділяється з-поміж таких альтернатив, як Snort, Zeek або Segan, завдяки підтримці багатопотокової роботи, протоколювання HTTP/TLS та інших корисних функцій.

Перед розгортанням Suricata має бути налаштована для таких змінних: який інтерфейс мережі використовувати, який діапазон IP-адрес буде ідентифікований як внутрішня мережа, і який діапазон IP-адресів слід вважати зовнішньою мережею.

Home_Net – ця змінна використовується для визначення внутрішньої мережі, яка має бути захищена.

External_Net використовується для визначення зовнішньої мережі.

Af-packet interface – змінна інтерфейсу в af-packet використовується для визначення мережевого інтерфейсу, який Suricata має використовувати для моніторингу.

2.2 Історія проекту Suricata

Проект стартував у 2009 р., а перший офіційний реліз відбувся в 2010 р..

Початкова мета проекту Suricata IDS полягала в розробці механізму виявлення вторгнень на основі сигнатур, схожого на його попередника Snort, але з іншими технологічними можливостями. Метою було побудувати NIDS, яка б використовувала ту ж мову виявлення, що й Snort, і зосередилася на спільноті. Ранні технологічні рішення полягали у впровадженні багатопоточності, розширеній підтримці HTTP і незалежному від порту розпізнаванні протоколу.

OISF, неприбуткова організація, наглядає за еволюцією платформи, створена для отримання коштів та піклування про сприяння та організацію зростання Suricata. Хронологія основних версій Suricata показана на рис. 2.2.

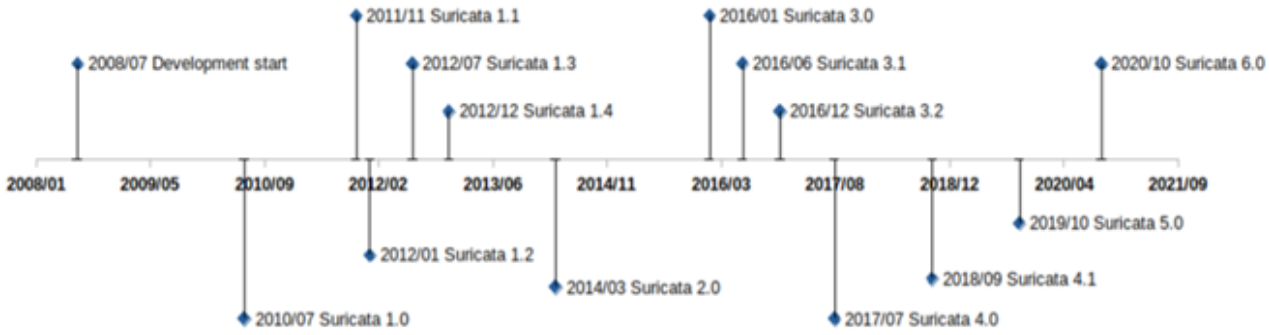


Рисунок 2.2 –Хронологія основних версій Suricata

Suricata 1.0.

Розуміння протоколу HTTP було, безумовно, найважливішим проривом у першому релізі.

Опублікована в липні 2010 р. після двох років розробки, змогла прочитати набір правил Snort, але також використовувала серію нових ключових слів для пошуку вмісту в полях протоколу HTTP за допомогою підходу, не залежного від портів.

Ще однією важливою особливістю Suricata 1.0 було розпізнавання протоколів. Двигун аналізує початок обміну в потоці, щоб дізнатися, який це протокол - повністю незалежний від порту рівня 4. Ця функція, що не залежить від портів, мала великий вплив на швидкість виявлення, оскільки багато шкідливих програм у той час використовували порт із високим номером для підключення до серверів командування та керування та HTTP для обміну інформацією.

Завдяки можливості знаходити HTTP незалежно від порту, що є великим досягненням, Suricata змогла точно виявити шкідливе програмне забезпечення.

Suricata також запропонувала багатоетапне виявлення завдяки включенню ключових слів, які були першим кроком до подолання низької виразності мови підписів, успадкованої від Snort.

Наприклад, сімейство ключових слів «flowbits» забезпечило спосіб передачі інформації між підписами і таким чином дозволити користувачам створити механізм стану. Хоча це було обмежено описом стану всередині єдиного потоку, це був справжній прогрес.

Ще одна особливість першого випуску порушила попереднє суворе визначення того, що таке IDS: реєстрація HTTP-запитів до файлу. Цього не було в початкових специфікаціях, але виявилося, що вона не надто складна у збірці і не мала начного впливу на продуктивність. Цей підхід продовжує визначати весь розвиток Suricata.

Suricata 1.2 (січень 2012 р.).

Інформація про транзакцію файлів була додана з версією 1.2 і була розширена у версії 1.3, реалізованій через шість місяців. Розуміння протоколу HTTP дало Suricata можливість побачити, що було передано в запитах, тому було логічним вилучення переданих файлів.

Це додалося у версії 1.2 разом із обчисленням контрольної суми файлу та журналом транзакцій. У Suricata 1.3 було додано ключове слово «filemd5», щоб перевірити, чи була контрольна сума md5 переданого файлу в списку, що зберігається у файлі. Пізніше ця функція буде розширена до sha1 і sha256 з ключовими словами «filesha1» і «filesha256».

Suricata 1.3 (липень 2012 р.).

Разом із Suricata 1.3 з'явилася підтримка TLS. Ця реалізація не включає дешифрування, а натомість є аналізом рукостискання TLS з вилученням унікальних характеристик транзакції, таких як тема сертифіката, емітент та його відбиток.

На цьому етапі зрозуміло, що Suricata відходить від класичної ролі IDS як презентатора простих даних. Система використовує складне декодування та вилучення даних, які просто не видно неозброєним оком. Це почалося з декомпресії повідомлення HTTP і продовжувалося з цього моменту.

Ця підтримка TLS тепер використовується для боротьби з еволюцією шкідливих програм, які почали використовувати зашифрований зв'язок. Наприклад, наявні на даний момент підписи тепер легко виявляють підключення до серверів із використанням конфігурацій OpenSSL за замовчуванням.

У цьому випуску також були додані виділені ключові слова TLS, і всі події TLS реєструються у спеціальному файлі.

Цей змішаний підхід – використання одночасно IDS і моніторингу безпеки мережі (NSM) – ґрунтується на тому, що було зроблено з HTTP. І надалі стане нормою. Тобто для кожного нового підтримуваного протоколу буде: додавання динамічної ідентифікації протоколу, реєстрація подій, пропозиція виділених ключових слів і розпакування файлів. Розвиток підтримки TLS тривав у кількох версіях Suricata.

Suricata 1.4 (грудень 2014 р.).

Suricata додала другу основну мову підпису на Lua, легкій, багатопарадигмовій мові програмування, розробленій переважно для вбудованого використання в додатках.

Тепер підписи могли включати сценарій Lua як функцію. Цей сценарій використовує доступні для Suricata буфери, такі як вміст пакету або інформація

TLS, і повертає значення 1 для збігу і 0 для відсутності збігу. Сценарій Lua також може створювати або змінювати flowbits змінні.

Завдяки цій додатковій можливості Suricata тепер мала справжню мову програмування, яку система могла використовувати для збереження станів. Це відкрило низку можливостей. Підтримка Lua, наприклад, могла бути використана для написання дуже точного підпису для виявлення спроб атаки Heartbleed. Фактично, цей підпис був доступний через кілька годин після оголошення атаки, і був би єдиним підхідом на основі сигнатур IDS, що забезпечував точне виявлення Heartbleed.

На жаль, підтримка Lua не мала того успіху, якого очікувала команда розробників – і з банальної причини. Для оцінки з підписом сценарій Lua для підпису має бути вставлений як файл поруч із файлом підписів. Але додавання цього типу файлів не підтримувалося існуючими інструментами керування підписами/правилами, і жодна велика організація з дослідження загроз не поширювала підписи за допомогою Lua з цієї простої причини. Інтерес до Lua існує і сьогодні, і посилення діяльності навколо інструментів керування підписами означає, що все ще є певна надія на підписи Lua.

Suricata 2.0 (березень 2014 р.).

Важливою віха в еволюції Suricata. Це сталося з додаванням JSON як бажаного формату для подій, створених Suricata. JSON забезпечив простий у розширенні та простий у використанні формат для всіх подій Suricata.

Завдяки форматуванню JSON було легко надіслати дані, згенеровані Suricata, до таких інструментів, як Elastic stack або Splunk. І вони поставлялися з вбудованою можливістю «кореляції», яку можна створити, використовуючи назву використовуваних полів. IP-адреса джерела завжди є полем «src_ip». Крім того, усі події тепер можна знайти в одному файлі (за замовчуванням), що містить різні типи журналів, попереджень та/або окремі транзакції DNS, SSH, TLS, HTTP, наприклад, і навіть дані про продуктивність.

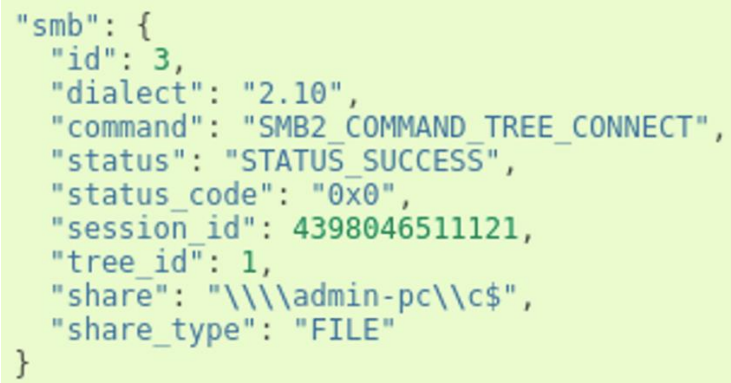
Suricata 3.0 (січень 2016).

Була опублікована в січні 2016 року, основною новою функцією було ключове слово «xbits». Концепція xbits полягає в тому, щоб вийти за межі обмежень

потокових біт, які не можна було використовувати в атаках з кількома потоками. Xbits – це еволюція потокових біт, у яких змінна приєднується до IP-адреси або до пари IP. Потім підписи можуть співпрацювати всередині кінцевого автомата, який не обмежується одним потоком.

Suricata 4.0 (липень 2017).

На додаток до підтримки низки нових протоколів, Suricata 4.0 представила в ядро більш безпечну та ефективну загальну техніку розбору. Використання комбінації мови Rust і парсера Nom заклало основу для швидкого збільшення кількості протоколів, які підтримує Suricata, без шкоди для безпеки та стабільності двигуна. Це виявиться критичним для відкриття шляху для повної функціональності NSM (рис. 2.3).



```
"smb": {
  "id": 3,
  "dialect": "2.10",
  "command": "SMB2_COMMAND_TREE_CONNECT",
  "status": "STATUS_SUCCESS",
  "status_code": "0x0",
  "session_id": 4398046511121,
  "tree_id": 1,
  "share": "\\\\admin-pc\\c$",
  "share_type": "FILE"
}
```


Рисунок 2.3 – Підоб'єкт SMB у події smb

Suricata 5 (жовтень 2019 р.).

Введення наборів даних додало можливість збігу в списку з більш ніж 50 різних буферів, а також звіряти список імен хостів з «відомо поганою» базою даних у імені хосту HTTP або в індикації імені сервера TLS.

Важливо, що ці списки можуть включати кілька елементів або мільйони з них, не погіршуючи продуктивність системи. Це ключова особливість, враховуючи тенденцію до обміну інформацією про загрози та використання таких інструментів, як MISP.

Ще одним дуже цікавим аспектом наборів даних є здатність Suricata додавати та видаляти елементи з набору за допомогою сигнатур, щоб ініціювати ці зміни.

Suricata 6 (жовтень 2020 р.).

Основним внеском Suricata 6 було розширення корпусу підтримуваних протоколів. З огляду на те, що майже половина з 10 мільйонів найкращих веб-сайтів підтримують протокол HTTP/2, для Suricata було важливо мати можливість реєструвати транзакції протоколу HTTP/2 і запускати на ньому виявлення загроз.

У цій версії також додана підтримка інших важливих протоколів, зокрема телеметричного транспорту черги повідомлень для середовищ Інтернету речей (IoT) та віддаленого буфера кадрів (використовується для сеансів віддаленого робочого столу).

Оскільки користувачі розгортають Suricata в середовищах зі швидкістю 100 Гбіт/с, а протоколювання прикладного рівня є важливою функцією, кількість подій, що генеруються в секунду, може бути досить високою. Наприклад, нерідкі випадки, коли розгортання зі швидкістю 100 Гбіт/с генерує сотні тисяч подій за секунду на одному зонді. Із Suricata 6 це було замінено користувацьким генератором JSON, написаним на Rust, що значно знижує навантаження на продуктивність журналу.

Suricata.

Спочатку створена як краща версія класичної системи виявлення вторгнень, Suricata з тих пір перетворилася на новий потужний клас інструментів безпеки мережі, забезпечуючи як сучасну систему виявлення вторгнень, так і систему моніторингу мережевої безпеки в єдиному високо продуктивному двигуні. Більше не потрібно розгортати дві окремі системи або навіть дві системи, інтегровані в одну. Suricata може виконувати обидві функції нативно.

Сьогодні ми бачимо постачальників інноваційних рішень, які розробили передові механізми виявлення інцидентів із збереженням стану та застосували машинне навчання поверх даних, наданих механізмом Suricata, щоб вирішити цю проблему. Але майбутні ітерації Suricata повинні будуть аналізувати більше внутрішніх протоколів, щоб залишатися актуальними.

2.3 Логи

Suricata постійно контролює мережу та генерує події на основі попереджень, аномалій, метаданих, інформації про файли та записів, що стосуються протоколу.

Логи як мережевих подій, так і працездатності suricata зберігаються в різних файлах і форматах у /var/log/suricata. Ці файли можна обробляти програмним забезпеченням сторонніх розробників для створення звітів і візуалізації, щоб краще зрозуміти, як поводиться Suricata.

EVE (extencible event format) – розширений формат події.

Основним файлом журналу подій є eve.json. це коли події зберігаються у форматі EVE.

Нижче наведено журнал EVE, який створює Suricata під час входу по SSH (рис. 2.4).


```
{
        "timestamp":"2019-10-07T23:37:32.964620+0000",
        "flow_id":1154589419900453,
        "in_iface":"eth0",
        "event_type":"ssh",
        "src_ip":"192.168.0.121",
        "src_port":47202,
        "dest_ip":"192.168.0.15",
        "dest_port":22,
        "proto":"TCP",
        "ssh":{
                "client":{
                        "proto_version":"2.0",
                        "software_version":"libssh-0.6.3"
                },
                "server":{
                        "proto_version":"2.0",
                        "software_version":"OpenSSH_7.6p1 Ubuntu-4ub
                }
        }
}
```


Рисунок 2.4 – Журнал EVE

Startup messages – повідомлення про запуск.

Сервісні повідомлення suricata, включаючи повідомлення про запуск та інші консольні повідомлення, зберігаються в suricata.log.

Fast.log.

Файл fast.log також містить записи для мережевих подій і сповіщень, наприклад eve.json. Однак це формати одного рядка для легкої інтеграції із поширеними утилітами *NIX, такими як grep та awk.

2.4 Правила Suricata

Правило складається з трьох частин: дії, заголовка та параметрів правила. Дія визначає, що станеться з пакетом, якщо він відповідає правилу:

- Pass – пропустіть пакунок, не створюючи сповіщення.
- Drop – якщо збігається, пакет буде негайно скинуто та записано в журнал.
- Reject – так само, як і при дії «відкидання», пакет буде негайно скинуто та записано в журнал, але і відправник, і одержувач отримають пакет відхилення.
- Alert – пакет дозволено пройти, але буде згенеровано сповіщення.

Заголовок повідомляє Suricata, для якого протоколу призначене правило, і визначає IP-адресу джерела та призначення. Що стосується протоколу, то існує три основні різновиди:

- TCP;
- UDP;
- ICMP.

Suricata також підтримує багато протоколів прикладного рівня, наприклад:

http, tls, ftp, dns, ssh, smtp та багато іншого.

Друга частина заголовка вказує IP-адресу та порт як джерела, так і призначення, структурований таким чином:

“IP джерела” “Порт джерела” -> “IP призначення” “Порт призначення”.

Також можна знайти відповідність в обох напрямках, замінивши -> на <> і написати "будь-який" замість діапазону або певної IP-адреси чи порту. Наступний рядок буде відповідати всім пакетам, які походять з IP-адреси $EXTERNAL_NET з будь-яким портом, спрямованим на IP-адресу $HOME_NET з портом 23.

$EXTERNAL_NET any -> $HOME_NET 23.

Остання частина правила містить параметри, записані парами ключ-значення або лише як ключі, якщо параметр є лише ключовим словом.

Параметри розділяються крапками з комою і повністю вкладені в дужки.

Найпоширеніші варіанти, які можна вважати базовими для написання правил:

msg – інформація щодо підпису та оповіщення;

sid – унікальний ідентифікаційний номер, присвоєний кожному правилу;

rev – представляє версію правила. Під час оновлення правила воно збільшується на 1.

Параметри правил, які складаються лише з msg, sid і rev, можуть виглядати так:

```
(msg:"Suspicious connection to port 20001"; sid:1252152; rev:1;)
```

Дія правила – це функціональна лінія поділу між IDS та IPS. IDS здатна лише ідентифікувати шкідливу поведінку, на відміну від IPS, який може як ідентифікувати, так і блокувати шкідливу поведінку, тим самим усуваючи мережі.

Примірник IDS повинен бути стратегічно розміщений в мережі, щоб він мав видимість для всієї мережевої активності, що походить зсередини та за межами мережі. Однак, коли справа доходить до налаштування IPS, окрім того, щоб Suricata працювала у вбудованому (IPS) режимі, також важливо переконатися, що IDS є точкою входу в мережу для інших хостів. Інакше Suricata не зможе скинути пакети.

3 ВСТАНОВЛЕННЯ ТА НАЛАШТУВАННЯ SURICATA

3.1 Системні вимоги

Мінімум 4 ГБ оперативної пам'яті та багатоядерний процесор для кращої продуктивності.

Встановлюю та налаштовую Suricata на Ubuntu 18.04 на віртуальну машину Virtual Box (рис. 3.1).

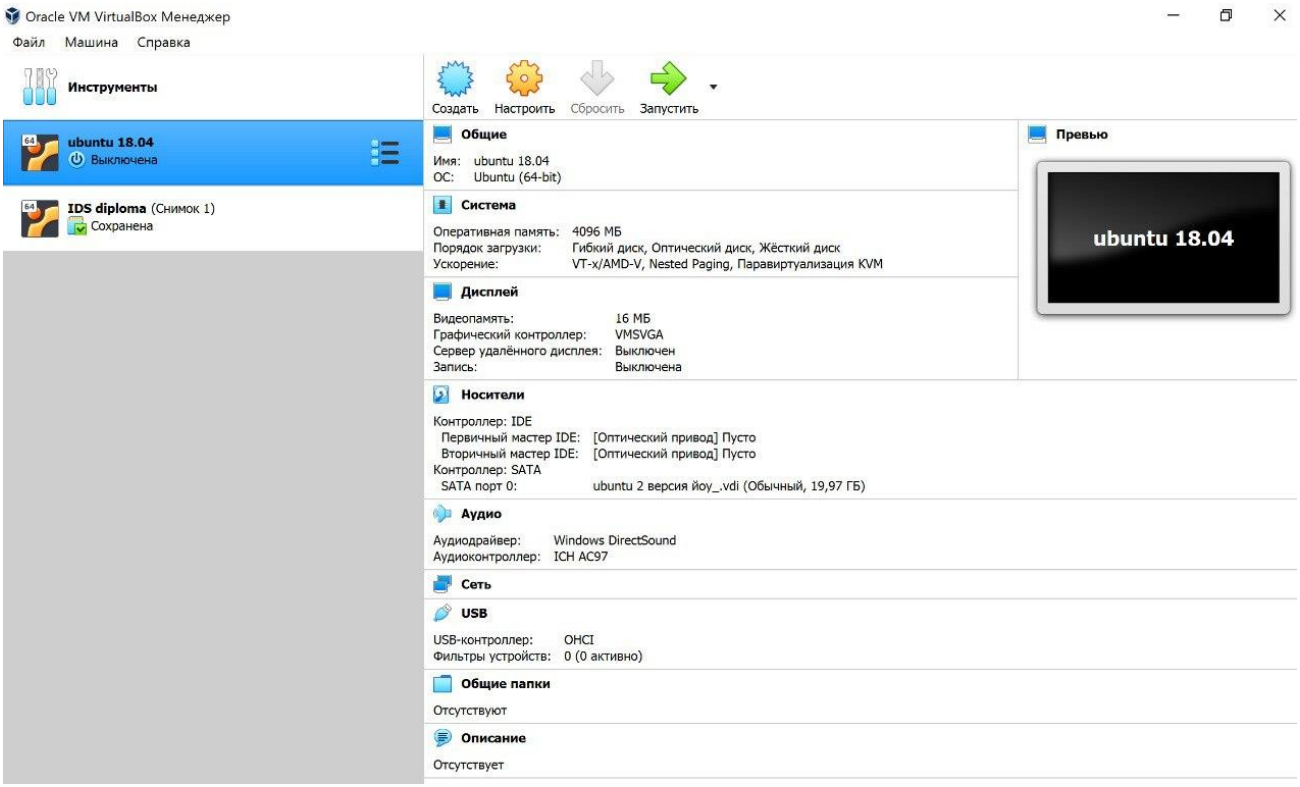


Рисунок 3.1 – Робота віртуальної машини Virtual Box

Існує два способи встановлення та налаштування Suricata в Ubuntu 18.04:

1. Встановлення з джерела.

2. Встановлення з репозиторію PPA.

3.2 Встановлення Suricata

```
sashalob@sashalob-VirtualBox:~$ sudo add-apt-repository ppa:oisf/suricata-stabl
e
[sudo] password for sashalob:
 Suricata IDS/IPS/NSM stable packages
https://suricata.io/
https://oisf.net/

Suricata IDS/IPS/NSM - Suricata is a high performance Intrusion Detection and P
revention System and Network Security Monitoring engine.

Open Source and owned by a community run non-profit foundation, the Open Inform
ation Security Foundation (OISF). Suricata is developed by the OISF, its suppor
ting vendors and the community.

This Engine supports:

- Multi-Threading - provides for extremely fast and flexible operation on multi
core systems.
- Multi Tenancy - Per vlan/Per interface
- Uses Rust for most protocol detection/parsing
- TLS/SSL certificate matching/logging
- JA3 TLS client fingerprinting
- JA3S TLS server fingerprinting
- IEEE 802.1ad (QinQ) and IEEE 802.1Q (VLAN) support
- VXLAN support
```

```
- All JSON output/logging capability
- IDS runmode
- IPS runmode
- IDPS runmode
- NSM runmode
- eBPF/XDP
- Automatic Protocol Detection and logging - IPv4/6, TCP, UDP, ICMP, HTTP, SMTP
, TLS, SSH, FTP, SMB, DNS, NFS, TFTP, KRB5, DHCP, IKEv2, SNMP, SIP, RDP
- SCADA automatic protocol detection - ENIP/DNP3/MODBUS
- File Extraction HTTP/SMTP/FTP/NFS/SMB - over 4000 file types recognized and e
xtracted from live traffic.
- File MD5/SHA1/SHA256 matching
- Gzip Decompression
- Fast IP Matching
- Datasets matching
- Rustlang enabled protocol detection
- Lua scripting
```

Додано спеціальний репозиторій PPA, і після оновлення індексу можна встановити Suricata.

```
sashalob@sashalob-VirtualBox:~$ sudo apt-get update && sudo apt-get install sur
icata
Hit:1 http://ua.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://ua.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://ua.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:5 http://ppa.launchpad.net/oisf/suricata-stable/ubuntu bionic InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libevent-core-2.1-6 libevent-pthreads-2.1-6 libhiredis0.13 libhtp2
  libhyperscan4 libluajit-5.1-2 libluajit-5.1-common liblzma-dev liblzma5
  libmaxminddb0 libnet1 libnetfilter-queue1
```

Після інсталяції Suricata я перевіряю, яка версія Suricata запущена і з якими параметрами, а також стан служби.

```
sashalob@sashalob-VirtualBox:~$ sudo suricata --build -info
[sudo] password for sashalob:
This is Suricata version 6.0.5 RELEASE
```

Встановлена версія 6.0.5.

```
sashalob@sashalob-VirtualBox:~$ sudo systemctl enable suricata.service
suricata.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
sashalob@sashalob-VirtualBox:~$ sudo systemctl status suricata
● suricata.service - LSB: Next Generation IDS/IPS
     Loaded: loaded (/etc/init.d/suricata; generated)
     Active: active (exited) since Thu 2022-06-02 17:13:14 EEST; 1min 11s ago
       Docs: man:systemd-sysv-generator(8)
      Tasks: 0 (limit: 4915)
     CGroup: /system.slice/suricata.service

чер 02 17:13:14 sashalob-VirtualBox systemd[1]: Starting LSB: Next Generation I
чер 02 17:13:14 sashalob-VirtualBox suricata[3980]: Starting suricata in IDS (a
чер 02 17:13:14 sashalob-VirtualBox systemd[1]: Started LSB: Next Generation ID
...skipping...
● suricata.service - LSB: Next Generation IDS/IPS
     Loaded: loaded (/etc/init.d/suricata; generated)
     Active: active (exited) since Thu 2022-06-02 17:13:14 EEST; 1min 11s ago
       Docs: man:systemd-sysv-generator(8)
      Tasks: 0 (limit: 4915)
     CGroup: /system.slice/suricata.service
```

Мені не потрібно, аби програма працювала під час налаштування, тож я зупиняю її командою:

```
sashalob@sashalob-VirtualBox:~$ sudo systemctl stop suricata
```

Переглядаючи контент директорії /etc/suricata/, можна помітити файл конфігурації – suricata.yaml, а також директорію rules, де міститься набір правил, який встановлюється разом із Suricata.

```
sashalob@sashalob-VirtualBox:~$ ls -al /etc/suricata
total 104
drwxr-xr-x   3 root root  4096 чер  2 17:13 .
drwxr-xr-x 125 root root 12288 чер  2 17:13 ..
-rw-r--r--   1 root root  3327 кві 21 11:08 classification.config
-rw-r--r--   1 root root  1375 кві 21 11:08 reference.config
drwxr-xr-x   2 root root  4096 чер  2 17:13 rules
-rw-r--r--   1 root root 73240 кві 22 09:53 suricata.yaml
-rw-r--r--   1 root root  1644 кві 21 11:08 threshold.config
```

Переглядаючи директорію, можна ознайомитись зі списком всіх правил різних типів, що мають в основі певні протоколи.

```
sashalob@sashalob-VirtualBox:~$ ls -al /etc/suricata/rules
total 136
drwxr-xr-x 2 root root  4096 чер  2 17:13 .
drwxr-xr-x 3 root root  4096 чер  2 17:13 ..
-rw-r--r-- 1 root root  1858 кві 21 11:08 app-layer-events.rules
-rw-r--r-- 1 root root 20821 кві 21 11:13 decoder-events.rules
-rw-r--r-- 1 root root   468 кві 21 11:08 dhcp-events.rules
-rw-r--r-- 1 root root  1221 кві 21 11:08 dnp3-events.rules
-rw-r--r-- 1 root root  1041 кві 21 11:08 dns-events.rules
-rw-r--r-- 1 root root  4003 кві 21 11:08 files.rules
-rw-r--r-- 1 root root  2128 кві 21 11:13 http2-events.rules
-rw-r--r-- 1 root root 13390 кві 21 11:13 http-events.rules
-rw-r--r-- 1 root root  2717 кві 21 11:13 ipsec-events.rules
-rw-r--r-- 1 root root   585 кві 21 11:08 kerberos-events.rules
-rw-r--r-- 1 root root  2078 кві 21 11:08 modbus-events.rules
-rw-r--r-- 1 root root  2013 кві 21 11:13 mqtt-events.rules
-rw-r--r-- 1 root root   558 кві 21 11:13 nfs-events.rules
-rw-r--r-- 1 root root   558 кві 21 11:08 ntp-events.rules
```

3.3 Базове налаштування

Визначаємо інтерфейс та IP-адресу, на яких Suricata має перевіряти мережеві пакети.

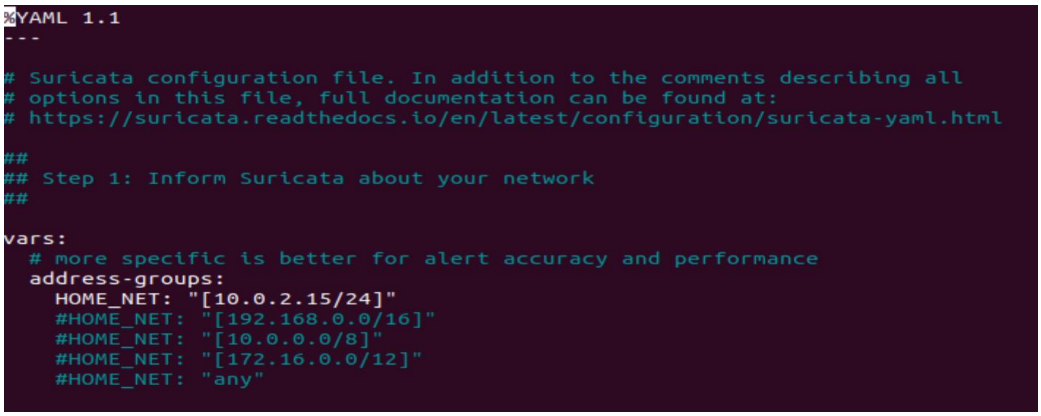
```
sashalob@sashalob-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a0ce:e128:c70:d3  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:cc:b9:da  txqueuelen 1000  (Ethernet)
        RX packets 8564  bytes 11847627 (11.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4520  bytes 345994 (345.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 317  bytes 27509 (27.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 317  bytes 27509 (27.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

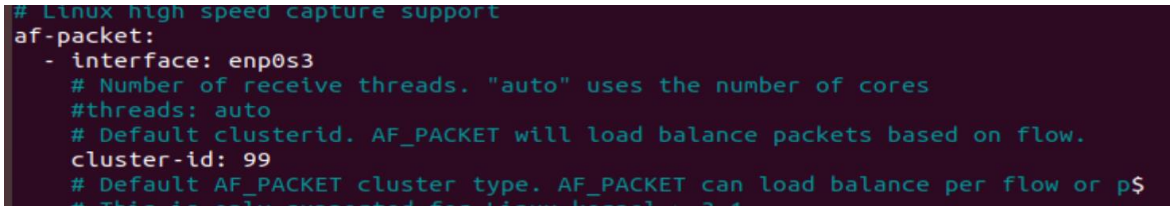
Бачимо, що мій інтерфейс enp0s3 і підмережа 10.0.2.15/24.

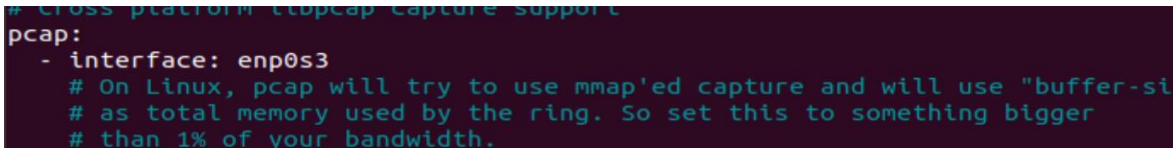
```
sashalob@sashalob-VirtualBox:~$ ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:cc:b9:da brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 85781sec preferred_lft 85781sec
    inet6 fe80::a0ce:e128:c70:d3/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Використовую цю інформацію, щоб налаштувати Suricata.

Аби зробити зміни в файлі /etc/suricata/suricata.yaml використовуємо редактор nano та права суперкористувача.

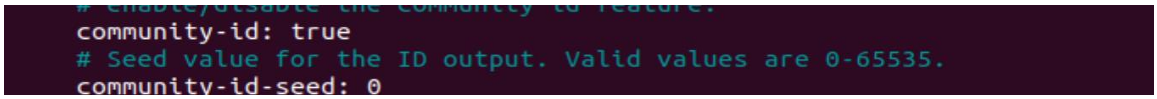






Suricata може включати поле ідентифікатора спільноти у вихідні дані JSON, щоб полегшити порівняння окремих записів подій із записами в наборах даних, створених іншими інструментами (інструментами, такими як Zeek або Elasticsearch).

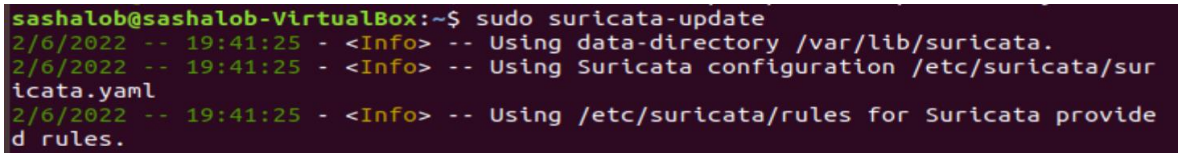
За замовчуванням Suricata створює файл журналу та зберігає журнали у форматах json. Тому я вмикаю функцію ідентифікації спільноти.

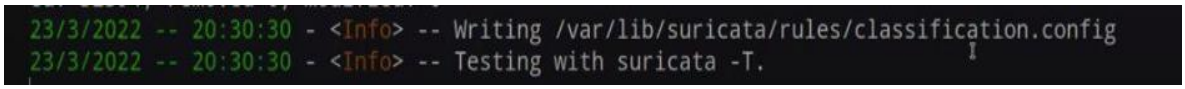


Ця конфігурація використовує найновіші рекомендовані налаштування для режиму роботи IDS для базових налаштувань.

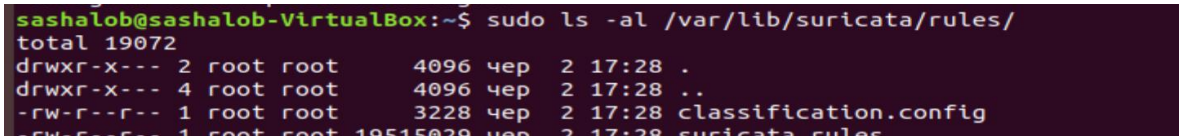
Наступне, що я змінюю, це rule-path. Я додаю файл користувацьких правил.

Необхідно запустити Suricata один раз, щоб зміни відобразилися.





Коли оновлення буде завершено, програма збирається виконати тест конфігурації Suricata, щоб переконатися, що немає нічого поганого в параметрах і синтаксисі, які я використовую.



Suricata пропонує можливість вказувати власні джерела, тому за замовчуванням вона дозволяє мені вибирати джерела, з яких я хотіла б отримати правила.

```
sashalob@sashalob-VirtualBox:~$ sudo suricata-update list-sources
2/6/2022 -- 17:30:29 - <Info> -- Using data-directory /var/lib/suricata.
2/6/2022 -- 17:30:29 - <Info> -- Using Suricata configuration /etc/suricata/sur
icata.yaml
2/6/2022 -- 17:30:29 - <Info> -- Using /etc/suricata/rules for Suricata provide
d rules.
2/6/2022 -- 17:30:29 - <Info> -- Found Suricata version 6.0.5 at /usr/bin/suric
ata.
2/6/2022 -- 17:30:29 - <Info> -- No source index found, running update-sources
2/6/2022 -- 17:30:29 - <Info> -- Downloading https://www.openinfosecfoundation.
org/rules/index.yaml
2/6/2022 -- 17:30:29 - <Info> -- Adding all sources
2/6/2022 -- 17:30:29 - <Info> -- Saved /var/lib/suricata/update/cache/index.yam
l
Name: et/open
  Vendor: Proofpoint
  Summary: Emerging Threats Open Ruleset
  License: MIT
```

Щоб додати або ввімкнути malsiro/win-malware.

```
sashalob@sashalob-VirtualBox:~$ sudo suricata-update enable-source malsilo/win-
malware
2/6/2022 -- 17:32:02 - <Info> -- Using data-directory /var/lib/suricata.
2/6/2022 -- 17:32:02 - <Info> -- Using Suricata configuration /etc/suricata/sur
icata.yaml
2/6/2022 -- 17:32:02 - <Info> -- Using /etc/suricata/rules for Suricata provide
d rules.
2/6/2022 -- 17:32:02 - <Info> -- Found Suricata version 6.0.5 at /usr/bin/suric
ata.
2/6/2022 -- 17:32:02 - <Info> -- Creating directory /var/lib/suricata/update/so
urces
2/6/2022 -- 17:32:02 - <Info> -- Enabling default source et/open
2/6/2022 -- 17:32:02 - <Info> -- Source malsilo/win-malware enabled
```

```
sashalob@sashalob-VirtualBox:~$ sudo suricata-update
2/6/2022 -- 17:32:19 - <Info> -- Using data-directory /var/lib/suricata.
2/6/2022 -- 17:32:19 - <Info> -- Using Suricata configuration /etc/suricata/sur
icata.yaml
```

```
sashalob@sashalob-VirtualBox:~$ sudo systemctl status suricata.service
```

```
● suricata.service - LSB: Next Generation IDS/IPS
     Loaded: loaded (/etc/init.d/suricata; generated)
     Active: failed (Result: exit-code) since Wed 2022-03-23 20:22:32 EDT; 12min ago
       Docs: man:systemd-sysv-generator(8)
    Process: 4300 ExecStop=/etc/init.d/suricata stop (code=exited, status=1/FAILURE)

Mar 23 20:21:17 blackbox systemd[1]: Starting LSB: Next Generation IDS/IPS...
Mar 23 20:21:17 blackbox suricata[3957]: Starting suricata in IDS (af-packet) mode... done.
Mar 23 20:21:17 blackbox systemd[1]: Started LSB: Next Generation IDS/IPS.
Mar 23 20:22:32 blackbox systemd[1]: Stopping LSB: Next Generation IDS/IPS...
Mar 23 20:22:32 blackbox suricata[4300]: Stopping suricata: /etc/init.d/suricata: 119: kill: No such process
Mar 23 20:22:32 blackbox systemd[1]: suricata.service: Control process exited, code=exited, status=1/FAILURE
Mar 23 20:22:32 blackbox systemd[1]: suricata.service: Failed with result 'exit-code'.
Mar 23 20:22:32 blackbox systemd[1]: Stopped LSB: Next Generation IDS/IPS.
```

Я хочу вказати власні правила та фактичний файл конфігурації, але спочатку мені потрібно запустити Suricata.

```
sashalob@sashalob-VirtualBox:~$ sudo systemctl start suricata.service
sashalob@sashalob-VirtualBox:~$ sudo systemctl status suricata.service
● suricata.service - LSB: Next Generation IDS/IPS
   Loaded: loaded (/etc/init.d/suricata; generated)
   Active: active (exited) since Thu 2022-06-02 17:13:14 EEST; 22min ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 4915)
   CGroup: /system.slice/suricata.service

чер 02 17:13:14 sashalob-VirtualBox systemd[1]: Starting LSB: Next Generation I
чер 02 17:13:14 sashalob-VirtualBox suricata[3980]: Starting suricata in IDS (a
чер 02 17:13:14 sashalob-VirtualBox systemd[1]: Started LSB: Next Generation ID
log file:
```

Активний статус означає, що Suricata фактично відстежує мережевий трафік і реєструє всі логи до каталогу.

```
sashalob@sashalob-VirtualBox:~$ ls -al /var/log/suricata
total 48
drwxr-xr-x  5 root root    4096 чер  2 17:13 .
drwxrwxr-x 12 root syslog  4096 чер  2 17:13 ..
drwxr-xr-x  2 root root    4096 кві 22 09:53 certs
drwxr-xr-x  2 root root    4096 кві 22 09:53 core
-rw-r--r--  1 root root       0 чер  2 17:13 eve.json
-rw-r--r--  1 root root       0 чер  2 17:13 fast.log
drwxr-xr-x  2 root root    4096 кві 22 09:53 files
-rw-r--r--  1 root root       0 чер  2 17:13 stats.log
-rw-r--r--  1 root root   22114 чер  2 17:34 suricata.log
```

Щоб запустити швидкий тест, я використовую одне з правил Suricata, яке включено у файл правил за замовчуванням.

```
sashalob@sashalob-VirtualBox:~$ curl http://testmynids.org/uid/index.html
uid=0(root) gid=0(root) groups=0(root)
```

```
sashalob@sashalob-VirtualBox:~$ sudo cat /var/log/suricata/fast.log
```

```
03/23/2022-20:38:59.426776  [**] [1:2013028:6] ET POLICY curl User-Agent Outbound [**] [Classification: Attempted Inform
n Leak] [Priority: 2] {TCP} 192.168.2.179:35286 -> 52.85.218.45:80
03/23/2022-20:38:59.487877  [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [**] [Classification: Potenti
Bad Traffic] [Priority: 2] {TCP} 52.85.218.45:80 -> 192.168.2.179:35286
```

Suricata працювала у фоновому режимі, і ми можемо побачити, де знаходиться фактичний журнал, звідки він надходить і куди йде.

Аби записати власні правила, я зупиняю Suricata.

```
sashalob@sashalob-VirtualBox:~$ sudo systemctl stop suricata.service
sashalob@sashalob-VirtualBox:~$ sudo nano /etc/suricata/rules/local.rules
```

```
alert icmp any any -> $HOME_NET any (msg:"ICMP Ping"; sid:1; rev:1;)
```

Тип правила – попередження, воно відстежуватиме все, що надходить із будь-якої зовнішньої мережі в мою мережу home, тому я вказую домашню змінну.

```
sashalob@sashalob-VirtualBox:~$ sudo suricata -T -c /etc/suricata/suricata.yaml
 -v
2/6/2022 -- 17:44:44 - <Info> - Running suricata under test mode
2/6/2022 -- 17:44:44 - <Notice> - This is Suricata version 6.0.5 RELEASE runnin
g in SYSTEM mode
```

```
sashalob@sashalob-VirtualBox:~$ sudo systemctl start suricata.service
```

Я запускаю ping із системи kali linux, і перевірю журнал, щоб побачити, чи було щось виявлено.

```
sashalob@sashalob-VirtualBox:~$ sudo cat /var/log/suricata/fast.log
```

```
> $ sudo cat /var/log/suricata/fast.log
03/23/2022-20:38:59.426776  [**] [1:2013028:6] ET POLICY curl User-Agent Outbound [**] [Classification: Attempted Inform
n Leak] [Priority: 2] {TCP} 192.168.2.179:35286 -> 52.85.218.45:80
03/23/2022-20:38:59.487877  [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [**] [Classification: Potentia
Bad Traffic] [Priority: 2] {TCP} 52.85.218.45:80 -> 192.168.2.179:35286
03/23/2022-20:44:16.584052  [**] [1:1:1] ICMP Ping [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.2.117:8 -
.168.2.179:0
03/23/2022-20:44:16.652363  [**] [1:1:1] ICMP Ping [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.2.179:0 -
.168.2.117:0
```

```
sashalob@sashalob-VirtualBox:~$ sudo tail -f /var/log/suricata/eve.json | jq 's
elect(.event_type=="alert")'
```

Дані, відображені у форматі json.


```
  "in_iface": "enp0s3",
  "event_type": "alert",
  "src_ip": "192.168.2.117",
  "src_port": 0,
  "dest_ip": "192.168.2.2",
  "dest_port": 0,
  "proto": "ICMP",
  "icmp_type": 8,
  "icmp_code": 0,
  "community_id": "1:S25lqIlGeYWkK61ZBIt+yHO27uk=",
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 1,
    "rev": 1,
    "signature": "ICMP Ping",
    "category": "",
    "severity": 3
  },
  "flow": {
    "pkts_toserver": 1,
    "pkts_toclient": 0,
    "bytes_toserver": 98,
    "bytes_toclient": 0,
    "start": "2022-03-23T20:48:19.625464-0400"
```

ВИСНОВКИ

Сьогодні корпоративні мережі обробляють все більше і більше трафіку, і багато з них зазвичай передають 10 гігабайт на секунду на магістралі.

Багатопотокова природа Suricata дозволяє користувачам масштабувати горизонтально на одному пристрої, додаючи потоки обробки пакетів у міру необхідності.

Suricata – чудовий інструмент, який можна мати у арсеналі виявлення вторгнень. Дані, отримані від Suricata, можуть допомогти створити географічну розбивку трафіку, який входить і виходить із мережі.

У цій роботі було розглянуто та встановлено систему виявлення атак Suricata.

Після встановлення Suricata було відредаговано конфігурацію за замовчуванням, щоб додати ідентифікатор потоку спільноти для використання з іншими інструментами безпеки, а також увімкнено перезавантаження правил в реальному часі та завантажили початковий набір правил.

Після перевірки конфігурації Suricata було запущено процес і створено тестовий HTTP-трафік, який підтвердив, що Suricata може виявити підозрілий трафік.

ПЕРЕЛІК ПОСИЛАНЬ

1. Approaches to Intrusion Detection and Prevention, 25 Aug, 2020. [Електронний ресурс] – Режим доступа: https://www.geeksforgeeks.org/approaches-to-intrusion-detection-and-prevention/?ref=lbp.

2. Intrusion Detection Systems: A Deep Dive Into NIDS & HIDS [Електронний ресурс] – Режим доступа: https://securityboulevard.com/2020/03/intrusion-detection-systems-a-deep-dive-into-nids-hids.

3. NIDs vs HIDs: Purpose, Core Functions & Benefits [Електронний ресурс] – Режим доступа: https://www.temok.com/blog/nids-vs-hids.

4. Compare Firewall and Intrusion Detection System (IDS) [Електронний ресурс] – Режим доступа: https://www.ques10.com/p/13428/compare-firewall-and-intrusion-detection-system-id.

5. IDS vs. IPS: What is the Difference? [Електронний ресурс] – Режим доступа: https://www.varonis.com/blog/ids-vs-ips.

6. Suricata: The First 12 Years of Innovation [Електронний ресурс] – Режим доступа: https://www.stamus-networks.com/blog/suricata-the-first-12-years-of-innovation.

7. How to Install Suricata NIDS on Ubuntu Linux [Електронний ресурс] – Режим доступа: https://www.rapid7.com/blog/post/2017/02/14/how-to-install-suricata-nids-on-ubuntu-linux.

8. How To Install Suricata on Ubuntu 20.04 [Електронний ресурс] – Режим доступа: https://www.digitalocean.com/community/tutorials/how-to-install-suricata-on-ubuntu-20-04.

9. Suricata User Guide [Електронний ресурс] – Режим доступа: https://suricata.readthedocs.io/en/suricata-6.0.0.